LLNL-TR-639492

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Application Co-design Plexus

J. Belak, S. Swaminarayan

June 18, 2013

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

5.8 Application Co-Design Plexus

Application-driven co-design is the process where scientific problem requirements influence computer architecture design, and node or system architecture technology constraints inform formulation and design of algorithms and software. The role and responsibility of the Co-design Plexus is to ensure timely coordination and communication of application co-design requirements and technical constraints between the Application Co-design Centers and the Extreme Scale Computing Effort (ESCE) ecosystem.

The application Co-design Plexus co-leads will organize a technical council consisting of the leads of all DOE ASCR and NNSA ASC Application Co-design Centers to:

- Collect and disseminate an evolving representation of the DOE application workload (list of proxy apps)
- Communicate the trade-off space within DOE applications
- Identify and communicate key research activities and challenges needed from other Nexus/Plexus within the ESCE (e.g. compilers, programming models, proxy architectures, simulators and tools, …)
- Identify and communicate tools for trade-off analysis with other Nexus/Plexus (e.g. architecture simulators and hack-a-thons with the Architecture Nexus)

5.8.1 Goals

Prepare future DOE applications for extreme-scale computing: The ultimate goal of the DOE ESCE is to create an effective extreme-scale simulation environment for DOE applications. The goal of the Co-design Plexus is to facilitate the preparation of DOE applications for extreme-scale computing and to provide a cogent description of areas where influence over future node and system architectures will have the greatest impact on the DOE portfolio of applications. The Co-design Plexus will also facilitate: coordination between current and future Application Co-design Centers; coordination between the Co-design Plexus and the ESCE Nexus/Plexus; and publicize the co-design lessons learned to the broader application community.

5.8.2 Strategy

Effective co-design must inject information where it can have impact, execute necessary follow-up, and record discussions leading to design decisions. While individual application and hardware/software co-design teams can and should have direct, sometimes confidential, contact with hardware and software vendors, the Application Co-design Plexus will facilitate and coordinate broader co-design activities. The Application Co-design Plexus will be open to all co-design centers, including current and future ASCR and NNSA/ASC co-design centers. The Application Co-design Plexus will lead the collective effort to:

- Establish a point of entry, through which any vendor or any participant in the ESCE can participate in the application-driven co-design process.
- Establish a repository (web page), with links to all active co-design efforts and related proxy app development efforts, through which the ESCE community can

gain access to the application requirements and the continuously evolving proxy application codes.

- Engage each of the ESCE Nexus/Plexus to ensure clear communication and common understanding,
- Coordinate activities (e.g., programming model research) and share resources (e.g., simulators, performance analysis tools, staff expertise) between co-design projects and ESCE Nexus/Plexus, and
- Assist in establishing and executing productive vehicles (e.g., open analysis results, workshops, on-site visits, secure on-line resources, etc.) for the rapid and multi-directional exchange of information within the ESCE ecosystem. The Application Co-design Plexus has recently co-organized "hack-a-thons" with the Architecture Nexus on architecture simulation and FastForward engagement. Hack-a-thons with the Performance Execution (performance modeling) and with the Software Stack (programming environments) Nexus are being planned.

5.8.3 Scope

Intra-Plexus Communication and Coordination

The Application Co-design Plexus provides intra-plexus communication and coordination among all projects in the plexus. These projects currently are (further detail below):

- Center for Exascale Simulation of Advanced Reactors (CESAR)
- Center for Exascale Simulation of Combustion in Turbulance (ExaCT)
- Exascale Co-Design Center for Materials in Extreme Environments (ExMatEX)
- National Nuclear Security Applications Co-Design Project (NSApp CDP)

Inter-Nexus/Plexus Communication and Coordination

The Application Co-design Plexus provides inter-nexus/plexus communication and coordination between the Application Co-design Centers and all Nexus/Plexus in the ESCE ecosystem and the Vendors. The funding of projects inside the Hardware and Software Nexus and the requirement for those projects to be relevant to the Co-design Centers' applications has kicked up the engagement a notch. The Co-design Centers provide a suite of proxy apps representing the DOE workload for trade-off analysis by the Centers and all Nexus/Plexus using simulators, models, and prototype hardware and software. The Co-design Centers use the results of this analysis to reformulate the proxy apps for emerging hardware and software design. Issues, challenges and expectations:

- **Applied Math Plexus:** All applications rely on numerical mathematics, e.g. linear algebra, fast fourier transforms, … It is not clear how current algorithms perform on emerging hardware concepts, let alone what new algorithms are required for power and resiliency. Challenge is matching proxies to concepts and vice versa, including proxies not yet authored.
- **Data Analysis Plexus:** The high cost of data movement and the enormity of data is driving a paradigm shift in data analysis, away from post simulation analysis towards *in situ* analysis and transforming data into an information-rich form to promote better understanding. Challenge is to identify interesting physics both

for storage/visualization and for physics refinement.

- **Resiliency Technical Council:** The dramatic increase in parallelism is expected to come with a dramatic increase in failure rates. The challenge is that application codes will need to have built in redundancy (micro-checkpointing) and the ability to migrate work to arbitrary nodes. In addition, there is a need for a programming model for resiliency.
- **Software Stack Nexus:** While all aspects of the software stack (tools, libraries, languages/compilers, resource/task management, runtime, communication, file system, energy/resilience management, operating system, etc.) are important for application performance, programming models stand out as most uncertain. It is through the programming model that the application developer thinks about the computer architecture. While the success of the MPI communication library suggests a path forward for inter-node parallelism, all of the new parallelism and complexity will be within the node. The challenge is that no consensus exists for a path forward to expose this parallelism to the application. Gaps: (1) MPI+X, what is X? (2) Language compilers lack the constructs and portability.
- **Data Management Nexus:** All applications perform I/O. Even on today's systems an optimum I/O strategy must be found. Gap: The optimum I/O strategy for the ESCE system is unknown. Challenge: How to use ESCE resources to perform data analysis during transit to I/O system?
- **Hardware Architecture Nexus:** Trade-off results from open simulation will be shared with all Nexus/Plexus in the ESCE. The challenge is extracting a set of hardware constraints based on vendor research while preserving Vendor IP. The Co-design Plexus expects the Hardware Architecture Nexus to provide Proxy Architectures and associated simulators for trade-off analysis. In this vein, the Co-design Plexus and Hardware Architecture Nexus have co-organized several deep-dive "hack-a-thons." The challenge is scaling this interaction to the *NxN* level to get all players involved.
- **Performance Execution Nexus:** Going forward, our performance analysis will have to include power and reliability. The Co-design Plexus anticipates the Performance Execution Nexus will provide a suite of modeling and simulation tools to assess this more general performance on Proxy Architectures (Hardware Architecture Nexus). While the Co-design Centers have used PIN and other tools to characterize single node behavior (e.g. to compare the computation workload of the proxy apps to their parents), the Co-design Plexus also anticipates the need for better tools to characterize the idioms of computational work and data movement needed for extreme scale computing.
- **System Engineering/Integration Nexus:** Siting the holistic ESCE machine offers some unique challenge to the applications such as the power management strategy and the relative allocation of resources. A challenge for the Co-design Plexus is to provide the System Engineering/Integration Nexus a "meta-workload" requirement for the ESCE system on the machine floor.

5.8.4 Research Activities and Outcomes

The Co-design Plexus consists of four research projects each using proxy apps to capture the requirements of their application domains and reformulating these proxy apps from the lessons learned from co-design trade-off analysis with other ESCE projects and Nexus/Plexus. These are:

DOE Office of Science ASCR Co-design Projects:

Center for Exascale Simulation of Advanced Reactors (CESAR)
Director: Andrew Siegel (ANL/UChicago)
Deputy Director: Paul Fischer (ANL)
Enable a coupled, next-generation nuclear reactor core simulation tool capable of efficient execution on exascale computing platforms.
http://cesar.mcs.anl.gov

Center for Exascale Simulation of Combustion in Turbulance (ExaCT)
Director: Jacqueline Chen (SNL)
Deputy Director: John Bell (LBNL)
Enable combustion scientists to perform first principles direct numerical simulations with sufficient physics fidelity to answer fundamental questions to meet pollutant and greenhouse gas emissions targets, reduce dependence on petroleum and promote economic competitiveness.
http://exactcodesign.org

Exascale Co-Design Center for Materials in Extreme Environments (ExMatEX)
Director: Tim Germann (LANL)
Deputy Director: Jim Belak (LLNL)
Establish the interrelationship between software and hardware required for materials simulation at the exascale while developing a multi-physics simulation framework for modeling materials subjected to extreme mechanical and radiation environments.
http://exmatex.lanl.gov

DOE NNSA ASC Co-design Project:

National Nuclear Security Applications Co-Design Project (NSApp CDP)

Project Leader: Sriram Swaminarayan (LANL)
Contacts: Sriram Swaminarayan (LANL), Rob Neely (LLNL), Mike Heroux (SNL-A)
The NSApp CDP provides the vehicle to engage in the collaborative process addressing the challenges of extreme scale computing, focusing on the unique aspects of integrated codes at the NNSA labs, including: emphasis on physics coupling in the ASC integrated codes; large codes that stress the software stack, whether it's tools (debuggers, hardware simulators, etc.), compilers, or operating systems; at-scale resiliency of the entire system; and the necessity to maintain the same code base for multiple platforms. Code reuse and performance portability is a high priority.
http://proxyapps.lanl.gov

Proxy Applications

The co-design centers make extensive use of proxy applications to represent application workflow and requirements both internally and to the broader ESCE ecosystem. In general, a small application code that proxifies (stands for) some aspect of the computational workflow of a full application is a proxy app. Proxy apps can be grouped into categories in increasing sophistication and fidelity to the parent applications. The current categories are:

- **Kernels:** standalone pieces of code that are small and performance- and tradeoff-impacting, even though decoupled from other application components.
- **Skeleton apps:** apps that reproduce the memory or communication patterns of a physics application or package, and make little or no attempt to investigate numerical performance.
- **Mini apps:** apps that combine some or all of the dominant numerical kernels contained in an actual stand-alone application and produce simplifications of physical phenomena.

Proxy apps are used both by the vendors and the rest of the ESCE ecosystem to understand the effects of hardware and software trade-offs, but also by co-design code team members to explore new technologies, languages, algorithms and programming models. It is important to emphasize that these proxy apps are not static, but evolve significantly during the co-design process. The co-design centers anticipate the requirement for domain application code-developers and hardware/software developers to spend significant together executing the co-design process. Additional details on proxy application, metrics for analysis, and co-design questions can be found in the ASCR and ASC whitepapers (ref 1,2).

5.8.5 High-level time estimate

(1) Deliver first suite of proxy apps for DOE workload (Fall 2012)
(2) Single point of entry / repository for lessons learned / Gap analysis / Challenges
(3) Coordinate engagement HW and SW communities, continuously, but there are mile posts defined by Fast Forward and other engagements
(4) Publish trade-off analyses (i.e. version 2 of proxy apps – Fall 2013, then version 3, …)
(5) Prototype extreme scale application (Fall 2016)
(6) Lessons learned in parent apps (this is an ongoing activity)

5.8.6 References

1. J. Belak, R. Rosner, J. Chen, T. Germann, 'Exascale Co-design Consortium (ECDC): Operations Plan', http://github.com/exmatex/docs, LLNL-AR-525077 (2012)
2. M. Hereoux, R. Neely, S. Swaminarayan , 'ASC Co-design Proxy App Strategy,' http://proxyapps.lanl.gov, LA-UR-13-20460 (2013)